Important Concepts from This Lecture

• Levels of Constraints
• Constraint Enforcement
• Update Propagation
• Viewing and Dropping Constraints
**Constraints**

- A constraint is a condition on the database that must hold at all times.
- There are various constraints on the database and they are enforced in different ways.
- There are at least three ways to enforce database constraints: Application level enforcement, database level enforcement, table level enforcement.

**Application-level Constraints**

- Application-level constraints that are enforced directly within the application and not in the database.
- Some constraints are difficult to enforced from within the database structure and leaves no choice but to enforce it at the application level.
  - A complex existence constraint must sometime be enforced at the application level. For example, if a married employee, whose spouse all works for the company, terminates their employment the dependents will not be removed because their existence also depends on the spouse being an employee.
- When a constraint is difficult or impossible to specify in full at the database or the table level, then it becomes necessary to enforce it at the application level.
- The trend is to capture “application behavior” within a database as much as possible.
Database-level Constraints

- Semantic Constraints
  - Enforced by TRIGGERS
  - Invoked when related to an INSERT, UPDATE, and/or DELETE occur on a table
  - Can execute before or after the modification and can affect other tables.
  - Example: Salary must not be decreased

- Existence Constraints
  - Enforced by a CASCADE, when changes are made to a strong entity association with a weak entity
  - When an employee terminates than the command also cascades to the dependency table via the part key.

Table-level Constraints

- Integrity Constraint
  - Rules that constrain the database such that it is always in a valid state as intended by the designer and associated requirements

- Security and Protection Constraints
  - Restrictions and authorization limitations that are applied to a database to protect it from misuse and unauthorized usage
Table-level Constraints

- **Primary Key Constraint**
  - Guarantees the Uniqueness and Minimality of the primary key of the table

- **Entity Constraint**
  - Enforcement that the primary key of the relation cannot be null

- **Domain Constraints** (range, min/max, enumerated values)
  - Enforced by the CHECK clause
  - Example: Age must be between 18 and 70

---

Table-level Constraints

- **Cardinality Constraints**
  - Cardinality constraints are restrictions that are placed on relationships in the data model, such that an instance of an entity can, through a relationship with another entity (or role within the same entity), be associated with one and only one instance(s) in another entity or role.

<table>
<thead>
<tr>
<th>1 Customer</th>
<th>1 Account</th>
<th>1 Customer</th>
<th>M Account</th>
<th>M Customer</th>
<th>M Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>A1</td>
<td>C1</td>
<td>A1</td>
<td>C1</td>
<td>A1</td>
</tr>
<tr>
<td>C2</td>
<td>A2</td>
<td>C2</td>
<td>A3</td>
<td>C2</td>
<td>A2</td>
</tr>
<tr>
<td>C3</td>
<td>A3</td>
<td>C3</td>
<td>A5</td>
<td>C2</td>
<td>A4</td>
</tr>
</tbody>
</table>

Dr. James Dullea, Constraint Enforcement & Referential Integrity Slide 7

Dr. James Dullea, Constraint Enforcement & Referential Integrity Slide 8
Table-level Constraints

Participation Constraints

- Participation constraints are restrictions (or the relaxing of a restriction) placed on the instances of an entity with respect to whether or not the instance needs to participate in the relationship with another entity (or role within an entity).

<table>
<thead>
<tr>
<th>Optional Customer</th>
<th>Mandatory Account</th>
<th>Optional Customer</th>
<th>Optional Account</th>
<th>Mandatory Customer</th>
<th>Mandatory Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>A1</td>
<td>C1</td>
<td>A1</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>A2</td>
<td>C2</td>
<td>A3</td>
<td>A2</td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>A3</td>
<td>C3</td>
<td>A5</td>
<td>A4</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>A4</td>
<td>C4</td>
<td>A5</td>
<td>A5</td>
<td></td>
</tr>
</tbody>
</table>

Table-level Constraints

- Domain Constraint
  - Restricts the values of an attribute

- Referential Integrity Constraint
  - The value of a Foreign Key, **if not null**, must exist in the referenced-to relation
  - A violation on an insert or update generates an error message and the row is not inserted or updated
Domain Constraints

DROP TABLE test_domain;
CREATE TABLE test_domain (  
    td_employee_id     VARCHAR2(9) NOT NULL,
    td_salary          NUMBER      NOT NULL,
    CONSTRAINT salary_ck CHECK
        (td_salary BETWEEN 10000 AND 100000),
    CONSTRAINT test_domain_pk
        PRIMARY KEY (td_employee_id));
INSERT INTO test_domain VALUES
    ('123456789',8000);
ERROR at line 1:  
ORA-02290: check constraint (CSC8490.SALARY_CK) violated

Constraint Enforcement

CREATE TABLE customer
    cnumber   VARCHAR2(8)   NOT NULL,
    ssn       VARCHAR2(11)  NOT NULL,
    name      VARCHAR2(30)  NOT NULL,
    zip       VARCHAR2(50)      NULL,
    CONSTRAINT customer_pk PRIMARY KEY (cnumber)
    CONSTRAINT customer_uq1 UNIQUE (ssn));
CREATE TABLE account (  
    anumber   VARCHAR2(20)  NOT NULL,
    balance   NUMBER(10,2)      NULL,
    CONSTRAINT account_pk PRIMARY KEY (anumber)

PRIMARY KEY AND UNIQUE automatically creates a unique index to enforce the uniqueness
Constraint Enforcement

Cardinality: One-to-One
CREATE TABLE ca (  
cnumber VARCHAR2(8) NOT NULL,  
anumber VARCHAR2(20) NOT NULL,  
CONSTRAINT ca_pk PRIMARY KEY (anumber),  
CONSTRAINT ca_uq1 UNIQUE (cnumber));

Cardinality: One-to-Many
CREATE TABLE ca (  
cnumber VARCHAR2(8) NOT NULL,  
anumber VARCHAR2(20) NOT NULL,  
CONSTRAINT ca_pk PRIMARY KEY (anumber));

Cardinality: Many-to-Many
CREATE TABLE ca (  
cnumber VARCHAR2(8) NOT NULL,  
anumber VARCHAR2(20) NOT NULL,  
CONSTRAINT ca_pk PRIMARY KEY (cnumber,anumber));

Common Terms in Database Systems

- Primary Key (PK) - uniquely identifies each tuple in the relation with no redundant attributes.
- Foreign Key (FK) - an attribute that is a PK in another relation.
- Composite Key - a PK consisting of multiple attributes
- Candidate Key (CK) - a set of attributes that can be used as a PK
- Alternate Key - a candidate key not designated as a PK
- Super Key - a set of attributes that includes a CK and may include redundant key attributes
- Prime Attribute - any attribute that belongs to a CK
- Secondary Key - an attribute that is used as an index
- Search Keys - An attribute used in a query
CHECK Clause, NOT NULL Constraint, PK & CKs

```sql
DROP TABLE employee;
CREATE TABLE employee (  
    ssn         VARCHAR2(11)  NOT NULL,
    fname VARCHAR2(20)  NOT NULL,
    lname VARCHAR2(20)  NOT NULL,
    sex          CHAR(1)           NULL,
    CONSTRAINT employee_ck1 CHECK (sex IN ('M','F')) ,
    CONSTRAINT employee_pk PRIMARY KEY (ssn),
    CONSTRAINT employee_uq1 UNIQUE (fname, lname));
```

Constraint Enforcement
Table needs to exist in order to reference it.

```sql
SQL> CREATE TABLE salespeople (  
    snum   varchar2(11) NOT NULL,
    sfname varchar2(15),
    slname varchar2(25),
    CONSTRAINT salespeople_pk PRIMARY KEY(snum));
Table created.

SQL> describe salespeople;
Name                            Null?    Type
------------------------------- -------- ----
SNUM NOT NULL VARCHAR2(11)
SFNAME VARCHAR2(15)
SLNAME VARCHAR2(25)
```
**Constraint Enforcement**

```
CREATE TABLE customers (
    cusnumber VARCHAR2(11) NOT NULL,
    cusname  VARCHAR2(30) NOT NULL,
    salesnumber VARCHAR2(11) NULL,
    CONSTRAINT customers_pk PRIMARY KEY(cusnumber),
    CONSTRAINT customers_uq1 UNIQUE(cusname),
    CONSTRAINT customers_fk1 FOREIGN KEY(salesnumber)
        REFERENCES salespeople(snum));
```

Table created.

SQL>
```
SQL> describe customers;
Name                            Null?    Type
------------------------------- -------- ----
CUSNUMBER NOT NULL VARCHAR2(11)
CUSNAME NOT NULL VARCHAR2(30)
SALESNUMBER VARCHAR2(11)
```

SQL> describe salespeople;
```
Name                            Null?    Type
------------------------------- -------- ----
SNUM NOT NULL VARCHAR2(11)
SFNAME                     VARCHAR2(15)
SLNAME                     VARCHAR2(25)
```

SQL>
```
Constraint Enforcement

SQL> INSERT INTO salespeople
2  VALUES ('605', 'James', 'Dullea');
1 row created.

SQL> INSERT INTO customers
2  VALUES ('101', 'Joe Buyer', '605');
1 row created.

SQL> INSERT INTO customers
2  VALUES ('102', 'Mary Purchaser', '666');
INSERT INTO customers
* 
ERROR at line 1:
OCA-30021: error preparing/executing SQL statement
[POL-5143] integrity constraint violation

SQL> ALTER TABLE customers ADD credit_lmt
decimal(10,2);
Table altered.

SQL> describe customers;
Name | Null? | Type
-----|-------|-----
CUSNUMBER | NOT NULL | VARCHAR2(11)
CUSNAME | NOT NULL | VARCHAR2(30)
SALESNUMBER | | VARCHAR2(11)
CREDIT_LMT | | NUMBER(10,2)
**Constraint Enforcement**

```sql
SQL> UPDATE customers SET credit_lmt = 10000.00
2  WHERE cusnumber = '101';
1 row updated.

SQL> SELECT * FROM customers;

<table>
<thead>
<tr>
<th>CUSNUMBER</th>
<th>CUSNAME</th>
<th>SALESNUMBER</th>
<th>CREDIT_LMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Joe Buyer</td>
<td>605</td>
<td>10000</td>
</tr>
</tbody>
</table>

SQL>
```

**Update Propagation**

- **Update Operations** - Insert, Delete, and Modify
  - Integrity constraints must not be violated by the update operations
- **Update Propagation**
  - Several update operations may have to be grouped together to achieve the desired result
  - Updates may propagate to cause other updates automatically that may be necessary to maintain integrity constraints
- **Update violations are handled by several strategies**
  - **Restrict** - cancel the operation causing the violation
  - **Cascade** - perform additional updates to correct the violation
  - **Trigger** - execute a user-specified error-correction routine
  - **Notify** - perform operation but inform user of the violation

See CREATE statement examples on page 215 of Elmasri for examples of "CASCADE"
**Restrict**

EMPLOYEE (E#, Ename, Address, Dept#)  
BORROW (Book#, E#, Due_date)  

An employee from EMPLOYEE table cannot be deleted if the E# appears in BORROW

If E# appears in BORROW, Then restrict (reject) Else delete.

Mechanisms: CHECK clause, TRIGGER

---

**Cascade**

Used in the case of mandatory participation - Existence Dependency (ED)

DEPARTMENT (Dept#, Dept_Name, Location)  
EMPLOYEE (E#, Ename, Address, Dept#)  
DEPENDENT (E#, ChildName, Age, Gender)  

If Dept# is changed in DEPARTMENT table, then Dept# in EMPLOYEE table must be changed
Deleting a department will require the reallocation of employees to other departments (mandatory participation)
Deleting an employee also deletes depending children (ED)

Mechanisms: ON DELETE CASCADE option in FOREIGN KEY clause
Trigger

Can be used in violations involving partial participation constraints

EMPLOYEE (E#, Ename, Address, Dept#)

If the Department is deleted, then Dept# in the EMPLOYEE table is set to null

Can be used in violations involving total participation constraints

If the Department is deleted, then Dept# in the EMPLOYEE table is set to a default value (such as 99)

Mechanisms: TRIGGER using PL/SQL

Dropping Tables with Constraint Enforcements

```
DROP TABLE table_a;
CREATE TABLE table_a  
  (a1 VARCHAR2(8),
   a2 VARCHAR2(8),
   a3 VARCHAR2(8),
   CONSTRAINT table_a_pk PRIMARY KEY(a1));

DROP TABLE table_b;
CREATE TABLE table_b  
  (b1 VARCHAR2(8),
   b2 VARCHAR2(8),
   b3 VARCHAR2(8),
   CONSTRAINT table_b_pk PRIMARY KEY(b1),
   CONSTRAINT table_b_fk1 FOREIGN KEY (b3)
     REFERENCES table_a(a1));

ALTER TABLE table_a ADD
  CONSTRAINT table_a_fk1 FOREIGN KEY (a3)
    REFERENCES table_b(b1);
```
Dropping Tables with Constraint Enforcements

DROP TABLE table_a;
DROP TABLE table_b;

ALTER TABLE table_a DROP CONSTRAINT table_a_fk1;
ALTER TABLE table_b DROP CONSTRAINT table_b_fk1;

DROP TABLE table_a;
DROP TABLE table_b;

DROP TABLE table_a CASCADE CONSTRAINTS;
DROP TABLE table_b CASCADE CONSTRAINTS;
### Viewing Constraint Enforcements

```sql
SELECT TABLE_NAME, COMMENTS
FROM DICT
WHERE TABLE_NAME LIKE 'USER_CONSTRAINTS';
```

```sql
SQL> SELECT TABLE_NAME, COMMENTS
2  FROM DICT
3  WHERE TABLE_NAME LIKE 'USER_CONSTRAINTS';
```

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_CONSTRAINTS</td>
<td>Constraint definitions on user's own tables</td>
</tr>
</tbody>
</table>

### Viewing Constraint Enforcements

```sql
SQL> DESCRIBE USER_CONSTRAINTS
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Null?</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>NOT NULL</td>
<td>VARCHAR2(30)</td>
</tr>
<tr>
<td>CONSTRAINT_NAME</td>
<td>NOT NULL</td>
<td>VARCHAR2(30)</td>
</tr>
<tr>
<td>CONSTRAINT_TYPE</td>
<td></td>
<td>VARCHAR2(1)</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>NOT NULL</td>
<td>VARCHAR2(30)</td>
</tr>
<tr>
<td>SEARCH_CONDITION</td>
<td></td>
<td>LONG</td>
</tr>
<tr>
<td>R_OWNER</td>
<td></td>
<td>VARCHAR2(30)</td>
</tr>
<tr>
<td>R_CONSTRAINT_NAME</td>
<td></td>
<td>VARCHAR2(30)</td>
</tr>
<tr>
<td>DELETE_RULE</td>
<td></td>
<td>VARCHAR2(9)</td>
</tr>
<tr>
<td>STATUS</td>
<td></td>
<td>VARCHAR2(8)</td>
</tr>
<tr>
<td>DEFERRABLE</td>
<td></td>
<td>VARCHAR2(14)</td>
</tr>
<tr>
<td>DEFERRED</td>
<td></td>
<td>VARCHAR2(9)</td>
</tr>
<tr>
<td>VALIDATED</td>
<td></td>
<td>VARCHAR2(13)</td>
</tr>
<tr>
<td>GENERATED</td>
<td></td>
<td>VARCHAR2(14)</td>
</tr>
<tr>
<td>BAD</td>
<td></td>
<td>VARCHAR2(3)</td>
</tr>
<tr>
<td>RELY</td>
<td></td>
<td>VARCHAR2(4)</td>
</tr>
<tr>
<td>LAST_CHANGE</td>
<td></td>
<td>DATE</td>
</tr>
</tbody>
</table>
Viewing Constraint Enforcements

SET PAGESIZE 66
COLUMN OWNER FORMAT A8
SELECT CONSTRAINT_NAME, TABLE_NAME, OWNER
FROM USER_CONSTRAINTS
WHERE CONSTRAINT_NAME NOT LIKE '%SYS%';

<table>
<thead>
<tr>
<th>CONSTRAINT_NAME</th>
<th>TABLE_NAME</th>
<th>OWNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMS_PK</td>
<td>CUSTOMERS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>DEPARTMENTS_PK</td>
<td>DEPARTMENTS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>EMPLOYEE_PK</td>
<td>EMPLOYER</td>
<td>DULLEA</td>
</tr>
<tr>
<td>EMPLOYEES_PK</td>
<td>EMPLOYERS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>ITEMS_PK</td>
<td>ITEMS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>JOBS_PK</td>
<td>JOBS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>LOCATIONS_PK</td>
<td>LOCATIONS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>PRICES_PK</td>
<td>PRICES</td>
<td>DULLEA</td>
</tr>
<tr>
<td>PRODUCTS_PK</td>
<td>PRODUCTS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>SALARY_GRADES_PK</td>
<td>SALARY_GRADES</td>
<td>DULLEA</td>
</tr>
<tr>
<td>SALES_ORDERS_PK</td>
<td>SALES_ORDERS</td>
<td>DULLEA</td>
</tr>
<tr>
<td>SCI_PK</td>
<td>STUDENT_COURSE_INSTRUCTOR</td>
<td>DULLEA</td>
</tr>
<tr>
<td>SUPPLIER_PART_AIRCRAFT_PK</td>
<td>SUPPLIER_PART_AIRCRAFT</td>
<td>DULLEA</td>
</tr>
<tr>
<td>TABLE_A_PK</td>
<td>TABLE_A</td>
<td>DULLEA</td>
</tr>
</tbody>
</table>

14 rows selected.

Assignment

SQL Assignment 4 - Constraints Enforcement